



# Environet™ Alert

## Public REST API

### User Manual

The information contained in this document is subject to change without notice and may not be suitable for all applications. While every precaution has been taken to ensure the accuracy and completeness of this document, Vertiv assumes no responsibility and disclaims all liability for damages result from use of this information or for any errors or omissions.

Refer to local regulations and building codes relating to the application, installation, and operation of this product. The consulting engineer, installer, and/or end user is responsible for compliance with all applicable laws and regulations relation to the application, installation, and operation of this product.

The products covered by this instruction manual are manufactured and/or sold by Vertiv. This document is the property of Vertiv and contains confidential and proprietary information owned by Vertiv. Any copying, use, or disclosure of it without the written permission of Vertiv is strictly prohibited.

Names of companies and products are trademarks or registered trademarks of the respective companies. Any questions regarding usage of trademark names should be directed to the original manufacturer.

### **Technical Support Site**

If you encounter any installation or operational issues with your product, check the pertinent section of this manual to see if the issue can be resolved by following outlined procedures.

Visit <https://www.vertiv.com/en-us/support/> for additional assistance.

## TABLE OF CONTENTS

<b>1 API Reference</b> .....	<b>1</b>
<b>2 Errors</b> .....	<b>3</b>
<b>3 Authentication</b> .....	<b>5</b>
<b>4 Querying Data</b> .....	<b>7</b>
4.1 \$filter .....	7
4.2 \$orderby .....	8
4.3 \$top and \$skip .....	8
<b>5 Analytics</b> .....	<b>9</b>
5.1 Analytics Categories .....	9
5.2 Querying for Data Sets .....	9
5.3 Data Set Entry Format .....	10
5.4 Category Specific Entries .....	13
5.5 Data Set Query Examples .....	20
<b>6 Analytics Metadata</b> .....	<b>25</b>
6.1 Query for Data Set Category Metadata .....	25

This page intentionally left blank

# 1 API Reference

The Vertiv™ Environet™ [REST](#) API provides rich access to data within the system to other management systems, data warehouses, mobile applications or any other system that is capable of web-based communications. Data contained within Vertiv Environet is available for retrieval and may also be created, updated or deleted where appropriate. Successful authentication is required for access to any data in the system. The API is versioned to allow backwards compatibility. Ensure the correct version is referenced when making calls to the server.

For the sake of simplicity, only relative URLs are shown in this guide. A fully functional URL will begin with the protocol, host and port that references the Environet server and will append a relative path for the desired resource.

For example, the authentication URL shown as */rest/v1/authentication* would in reality be something like <http://192.168.0.1/rest/v1/authentication>, where *192.168.0.1* would be the actual IP address of the Environet server.

This page intentionally left blank

## 2 Errors

The API uses conventional HTTP response codes to indicate the success or failure of an API request. In general: Codes in the 2xx range indicate success. Codes in the 4xx range indicate an error that failed given the information provided (e.g., authentication required, a required parameter was omitted, etc.). Codes in the 5xx range indicate an error with the Environet server. Technical support should be notified of a 5xx error so it can be addressed.

**Table 2.1 HTTPS Status Codes**

Code	Description
200 - OK	The request was successful and response data was returned.
201 - Created	The request was successful in creating the resource.
204 - No Content	The request was successful and there is no data to return.
400 - Bad Request	The request was unacceptable, often due to malformed syntax.
403 - Forbidden	Authentication is required before the server will process the request.
404 - Not Found	The server does not recognize the URL resource that was requested.
409 - Conflict	The end point was called in an unexpected way or the data sent is invalid.
415 - Unsupported Media Type	The end point cannot return the media type requested.
500 - Internal Server Error	An error occurred on the server.
501 - Not Implemented	The REST end point has not been implemented.

This page intentionally left blank

## 3 Authentication

Before data can be accessed using the Vertiv™ Environet™ [REST](#) API, the client must authenticate. Successful authentication requires an HTTP POST of *x-www-form-urlencoded* credentials to the authentication endpoint with a valid username and password. Upon successful authentication, a session cookie will be returned to the client that must be provided with subsequent requests. The cookie expires and is no longer valid when the client session ends.

**Method:** *POST*

**URL:** */rest/v1/authentication*

**Content-type:** *application/x-www-form-urlencoded*

### Example HTTP Request

```
$.ajax({  
  method : "POST",  
  url: "/rest/v1/authentication",  
  data: {username: "user", password: "pass" }  
});
```

### Example HTTP Response

```
204 No Content
```

This page intentionally left blank

## 4 Querying Data

The Vertiv™ Environet™ [REST](#) API provides a query system to control the way the data is processed. This is done using query string parameters to filter, order and page the data.

### 4.1 \$filter

The *\$filter* parameter allows selecting a subset of data using logical operators. All operators take a field and most use one or more values. Operators can be grouped using parenthesis and each group can be joined using *and* or *or*. Filterable fields are any top-level field (key in JSON) that is returned in the data that is of a following type: number, Boolean (true, false), date, or text. Values to test against can be of a following types: number, Boolean, date or text. Dates must be formatted using the *ISO 8601 date/time* format and contained in single quotes. Any text must also be contained in single quotes.

**NOTE: Filtering is case insensitive.**

#### Examples

*# Select all data where the 'name' field is "steve" and the 'age' field is a number greater than or equal to 21# Also select all data where the 'admin' field is true*

`$filter=(name eq 'steve' and age ge 21) or admin eq true`

*# Select all data where the 'startDate' field is a date between January 1st 2018 and February 1st 2018*

`$filter=between(startDate,'2018-1-1T00:00:00Z','2018-2-1T00:00:00Z')`

**Table 4.1 Filtering Parameters**

Operator	Example	Description
eq	name eq 'john'	Name is "john".
ne	age ne 18	Age is not 18.
gt	age gt 21	Age is greater than 21.
ge	age ge 21	Age is greater than or equal to 21.
lt	age lt 21	Age is less than 21.
le	age le 21	Age is less than or equal to 21.
contains	contains(name,'oh')	Name contains "oh".
startswith	startswith(name,'st')	Name starts with "st".
endswith	endswith(name,'n')	Name ends with "n".
between	between(age,18,21)	Age is between 18 and 21 (inclusive).
in	in(name;'john','steve')	Name is one of the values "john" or "steve".
isnull	isnull(name)	Name does not have a value.
isnotnull	isnotnull(name)	Name has a value.

## 4.2 \$orderby

The *\$orderby* parameter allows the data to be sorted by a field in either ascending or descending order. Sorting is case-sensitive and alphanumeric allowing text with numeric prefixes and suffixes to be sorted logically.

### Examples

*# Order data by the 'age' field in ascending (default) order*

*\$orderby=age*

*# Order data by the 'name' field alphabetically in descending order*

*\$orderby=name desc*

## 4.3 \$stop and \$skip

The *\$stop* and *\$skip* parameters allow a paged set of data to be returned. *\$stop* controls how many records to return. By default, all results are returned. *\$skip* controls which record to start with..

### Examples

*# Get the first page of 10 results (\$skip=0 is inferred)*

*\$stop=10*

*# Get the second page of 10 results*

*\$stop=10&\$skip=10*

## 5 Analytics

Allows querying for historical data sets across different object types in the system. Object types are represented by data set categories. Each data set represents a single object from a specific category and contains some metadata about the object. Each data set category makes several entries that pertain to that category of data available for querying.

Method	URL	Content-Type	Description
GET	/rest/v1/data-sets/{category}	application/json	Get sets of data for a data set category.

### 5.1 Analytics Categories

Table 5.1 Analytics Categories

Category	Description
GROUP	A hierarchical organization of devices.
LOCATION	A geographical or functional location (aka, site, facility, room, etc.).
RACK	A rack (aka, cabinet) within a site.
ASSET	An asset within a rack
DEVICE	A logical grouping of data points, usually corresponding to a physical piece of equipment.
CIRCUIT	A logical grouping of electrical data points, usually corresponding to a physical piece of equipment.
TENANT	A customer, department or business group that operates and maintains equipment in the data center
POINT	A data point of a device, rack, circuit or group.
MAINTENANCE	A maintenance event in the system; may be scheduled or unscheduled.
ALARM	An off-normal condition in the system.

### 5.2 Querying for Data Sets

The analytics API allows for querying using the standard system for [Querying Data](#) on page 7. Every data set has a set of common fields that can be used for querying for data sets. Each category of data set adds additional fields to allow for more fine-grained control over querying that category of data.

**NOTE:** In previous implementations of the API, data sets could be queried using the *categoryId* field or using *selectors*. The use of the *categoryId* field and *selectors* is deprecated in favor of the new query system.

#### Base Data Set Entity

```
{
  "category": String,
  "categoryId": String (Unique Identifier),
  "displayName": String,
  "path": String,
  "status": {
    "overall": String,
    "alarmCount": Number,
    "warningCount": Number,
  }
}
```

```

    "infoCount": Number,
    "downCount": Number,
    "faultCount": Number,
    "maintenanceCount": Number,
    "disabledCount": Number,
    "normalCount": Number,
    "unknownCount": Number
  },
  "links": [
    {
      "rel": String,
      "href": String
    }
  ],
  "entries": List of Entry Entities
}

```

**Table 5.2 Additional Fields**

Additional Fields	Group	Location	Rack	Asset	Device	Circuit	Tenant	Point	Maintenance	Alarm
siteName		X	X	X	X	X		X		X
groupPath	X							X		X
deviceName					X	X		X		X
circuitName						X		X		X
rackName			X	X		X		X		X
assetName				X						
assetEnclosureName				X						
tenantName			X			X	X	X		X
tenantId			X			X	X	X		X
pointName								X		
maintenanceName									X	
alarmSource										X
alarmTime										X

### 5.3 Data Set Entry Format

For the requested data sets, a list of entry selectors can be used to query specific information about each data set returned.

Entries determine what properties of the category the data sets will contain. In the data set endpoint URL, the *ent* query parameter defines entry selectors.

**Entry Entity**

```

{
  "name": String,
  "label": String,

```

```

    "units": String - present if applicable,
    "status": String - present if entry represents a point that reports status,
    "pointId": String - present if entry represents a point,
    "alarmLowLimit": String - present if entry represents a point that has alarm
thresholds,
    "alarmHighLimit": String - present if entry represents a point that has alarm
thresholds,
    "warnLowLimit": String - present if entry represents a point that has warning
thresholds,
    "warnHighLimit": String - present if entry represents a point that has warning
thresholds,
    "values": [
      {
        "value": String,
        "timestamp": ISO 8601 date/time
      }
    ]
  }
}

```

## Entry Selector Format

Each entry selector consists of nine fields, delimited by commas, that must be specified in the order shown below. Entry label and entry name are the only required fields; every other field is optional. However, if any other field is used, then all fields before it must be specified, even if empty, in order to ensure proper processing of entry selectors. Multiple entry selectors are specified using a semicolon to delineate each selector.

```
ent= {entry label},{entry name},{rollup},{rollup function},{timeframe},{timeframe start},
{timeframe end},{hidden},{calculation function};[next selector...]
```

**Table 5.3 Entry Selectors**

Category	Default	Description
Entry Label		Identifier to uniquely identify the entry returned by the selector.
Entry Name		The name of the entry being requested.
Rollup	NONE	The timeframe used to rollup historical values: <i>NONE, ONE_MIN, FIVE_MIN, FIFTEEN_MIN, THIRTY_MIN, HOURLY, DAILY, WEEKLY, MONTHLY, YEARLY, ALL.</i>
Rollup Function		The function used to apply the rollup (required if rollup is not <i>NONE</i> ): <i>AVERAGE, MINIMUM, MAXIMUM, SUM, NET_CONSUMPTION.</i>
Timeframe	NOW	The timeframe to get historical values for are: <i>NOW, TODAY, YESTERDAY, LAST_WEEK, LAST_MONTH, LAST_YEAR, WEEK_TO_DATE, MONTH_TO_DATE, YEAR_TO_DATE, CURRENT_BILLING_CYCLE, LAST_BILLING_CYCLE, ALL, USER_DEFINED.</i>
Timeframe Start		<i>ISO 8601 datetime</i> used when the timeframe is set to <i>CURRENT_BILLING_CYCLE, LAST_BILLING_CYCLE</i> or <i>USER_DEFINED</i> to indicate the start of the timeframe to query or the time and day of the month the billing cycle starts on.
Timeframe End		<i>ISO 8601 datetime</i> used when the timeframe is set to <i>USER_DEFINED</i> to indicate the end of the timeframe to query.
Hidden	False	If set to <i>"true"</i> , the entry will not be returned in the data sets, but it can be used in calculations.
Calculation		A function for defining a calculated entry. The calculated entry can use values from any other entry defined by the selectors for the query in its calculation.

For aggregate rollup functions (*Avg, Sum, Net Consumption*) with a rollup specified, the timestamp on each rolled-up value returned will be the end date of the rollup timeframe requested. For minimum and maximum, the timestamp will be the timestamp of the minimum or maximum value recorded in the rollup interval. If there is no rollup, the timestamp will be the timestamp the value was recorded at. When there is no value for a given timeframe or rollup interval, no value is returned.

### Calculated Entries

Calculated values are defined using mathematical operations. Most operations accept multiple operands (inputs), which may be numeric constants or values from another entry, including other calculated values.

All entries, including calculated entries, have a user-definable label that can be used in other calculated entries. For example, a user may want to call a calculated entry *Average Voltage* or *MaxTempDeviation*.

To use values from other entries as inputs, reference the entry labels. For example, a three-phase circuit may have separate entries for each line-to-line voltage. Each entry can be given any user label. For example, *VoltsAB*, *VoltsBC*, and *VoltsCA* and then used in calculations:

```
# Average line-to-line voltage:
AVG(VoltsAB,VoltsBC,VoltsCA)
```

```
# Operations can be nested
# Maximum deviation from the average of any line-to-line voltage:
MAX(ABS(SUB(VoltsAB,AvgVoltage)),ABS(SUB(VoltsBC,AvgVoltage)),ABS(SUB(VoltsCA,AvgVoltage)))
```

```
# Numeric constant inputs may be combined with entry inputs
# % voltage imbalance
MULT(DIV(MaxDeviation,AvgVoltage),100)
```

**Table 5.4 Supported Calculation Operations**

Operation	Description	Example
<i>SUM</i>	Adds inputs.	$SUM(1,2,3) = (1 + 2 + 3) = 6$
<i>SUB</i>	Subtracts inputs in left-to-right sequence.	$SUB(1,2,3) = ((1 - 2) - 3) = -4$
<i>MULT</i>	Multiplies inputs.	$MULT(1,2,3) = (1 \times 2 \times 3) = 6$
<i>DIV</i>	Divides first input by the subsequent inputs in left-to-right sequence.	$DIV(8,2,2) = ((8 / 2) / 2) = 2$
<i>MIN</i>	Returns the minimum value among the inputs.	$MIN(3,1,2) = 1$
<i>MAX</i>	Returns the maximum value among the inputs.	$MAX(3,1,2) = 3$
<i>AVG</i>	Returns the average of the inputs.	$AVG(1,2,3) = 2$
<i>ABS</i>	Returns the absolute value of the input.	$ABS(-4) = 4$
<i>ROUND</i>	Rounds the first input to the number of decimal places specified by the second input or 0 decimal places.	$ROUND(1.56,1) = 1.6$ , $ROUND(1.56) = 2$
<i>ALIAS</i>	Groups analogous entries that have different names into a single alias entry.	$ALIAS(Return\ Temp,Return\ Temperature) = first\ found\ value\ from\ each\ entry\ for\ a\ data\ set$

## 5.4 Category Specific Entries

Each Data Set Category provides its own set of entries that can be requested by name using entry selectors.

**Table 5.5 Group Entries**

Entry Name	Description
NAME	Name of the site..
PATH	The path of the group within the hierarchy.
STATUS	Status.
LOCATION	Site name.
KPI1	The values for the point assigned as KPI 1.
KPI2	The values for the point assigned as KPI 2.
KPI3	The values for the point assigned as KPI 3.

**NOTE: If the entry name requested in the selector does not match a name above and a point under that group exists with the same name, the value of the point will be returned.**

**Table 5.6 Location Entries**

Entry Name	Description
NAME	Name of the site.
RACKS	Collection of racks represented by <i>siteName/rackName</i> .
DEVICES	Collection of devices represented by <i>siteName/deviceName</i> .
TENANTS	Collection of tenants represented by <i>tenantName</i>
RACK_COUNT	Number of racks at this site.
DEVICE_COUNT	Number of devices at this site.
TENANT_COUNT	Number of tenants this site is associated to

**Table 5.7 Rack Entries**

Entry Name	Description
NAME	Name of the rack
STATUS	Status
LOCATION	Site name
DEVICES	Collection of devices that contain circuits associated to the rack represented by <i>siteName/deviceName</i>
CIRCUITS	Collection of circuits associated to the rack represented by <i>siteName/deviceName/circuitName</i>
ASSET	Collection of assets placed within the rack represented by <i>assetName</i>
ASSET_COUNT	Number of assets placed within the rack
TENANT_NAME	Collection of tenants linked to circuits associated to the rack represented by <i>tenantName</i>
TENANT_ID	Collection of tenants linked to circuits associated to the rack represented by <i>tenantId</i>
RACK_U_FRONT	RU Front
RACK_U_REAR	RU Rear

**Table 5.7 Rack Entries (continued)**

Entry Name	Description
RU_FREE_FRONT	Available RU Front
RU_FREE_REAR	Available RU Rear
MAX_CONT_FREE_RU_FRONT	Max Contiguous RU Front
MAX_CONT_FREE_RU_REAR	Max Contiguous RU Rear
TOTAL_KW	Total KW of circuits associated to the rack
TOTAL_KW_CAPACITY	Total KW Capacity
KW_CAPACITY_PERCENT	KW Capacity percent
TOTAL_FACEPLATE_POWER	Total faceplate power of assets placed within the rack
TEMPERATURE_MAX	Temperature Max
TEMPERATURE_TOP	Temperature Top
TEMPERATURE_MIDDLE	Temperature Middle
TEMPERATURE_BOTTOM	Temperature Bottom
TEMPERATURE_AVG	Temperature Average
KPI1	The values for the point assigned as KPI 1
KPI2	The values for the point assigned as KPI 2
KPI3	The values for the point assigned as KPI 3

**NOTE:** If the entry name requested in the selector does not match a name above and a custom field by that name exists, it will return the value of the custom field. Otherwise, if a point under that rack exists with the same name, the value of the point will be returned.

**Table 5.8 Asset Entries**

Entry Name	Description
NAME	Name of the asset
LOCATION	Site name if the asset is placed in a rack
RACK	Rack the asset is placed in represented by siteName/rackName
POSITION	RU Position within the rack
ENCLOSURE	The parent asset enclosure if placed in an enclosure
ENCLOSURE_SLOT	The slot number representing the placement within the enclosure
ENCLOSURE_ASSETS	Collection of assets placed within this asset enclosure
TOTAL_SLOTS	Total number of slots in this asset enclosure
AVAILABLE_SLOTS	Available number of slots in this asset enclosure
MAKE	Make
MODEL	Model
SERIAL_NUMBER	Serial Number

**Table 5.8 Asset Entries (continued)**

Entry Name	Description
ASSET_TAG	Asset tag
INSTALL_DATE	Installation date
WARRANTY_EXPIRATION	Warranty expiration date

**Table 5.9 Device Entries**

Entry Name	Description
NAME	Name of the device
STATUS	Status
DEVICE_TYPE	Device Type
LOCATION	Site name
RACKS	Collection of racks associated to circuits under this device represented by siteName/rackName
RACK_COUNT	Number of racks associated to circuits under this device
ALARM_COUNT	Number of alarms for the device
IP_ADDRESS	IP Address
CIRCUIT_NUMBERING	The circuit enclosure's number scheme if the device contains a circuit enclosure
PANEL_SIDES	The circuit enclosure's number of sides if the device contains a circuit enclosure
TERMINALS_PER_SIDE	The circuit enclosure's number of terminals per side if the device contains a circuit enclosure
KPI1	The values for the point assigned as KPI 1
KPI2	The values for the point assigned as KPI 2
KPI3	The values for the point assigned as KPI 3

**NOTE:** If the entry name requested in the selector does not match a name above and a point under that device exists with the same name, the value of the point will be returned. Points under circuits of that device can also be referenced by using the circuit name as a prefix.

**Table 5.10 Circuit Entries**

Entry Name	Description
NAME	Name of the circuit.
STATUS	Status.
LOCATION	Site name of the device.
DEVICE	Device the circuit is under represented by <i>siteName/deviceName</i> .
RACK	Rack that the circuit is associated to represented by <i>siteName/rackName</i> .
FEED	The feed that is associated to the rack.
TENANT_NAME	The name of the tenant linked to the circuit represented by <i>tenantName</i>
TENANT_ID	The name of the tenant linked to the circuit represented by <i>tenantId</i>
CIRCUIT_TYPE	Single-phase wye (120V); single-phase delta (208V); three-phase delta or three-phase wye.
PHASE	Phase.

**Table 5.10 Circuit Entries (continued)**

Entry Name	Description
DIRECTION	Input or output.
CHANNEL	The channel label.
POLES	The number of phases of the circuit: 1 for single-phase wye (120V), 2 for single-phase delta (208V), 3 for three-phase delta or three-phase wye.
AMPS	Single-phase RMS current for single-phase circuits and three-phase current for three-phase circuits.
VOLTS_LN	The LN voltage of the circuit representing single-phase or three-phase voltage, depending on type.
VOLTS_LL	The LL voltage of the circuit representing single-phase or three-phase voltage, depending on type.
BREAKER_SIZE	Breaker size.
KW	Total real power delivered by the circuit regardless of type.
KWH	Energy consumed by devices connected to the circuit regardless of type.
KVA	Total apparent power delivered by the circuit regardless of type.
POWER_FACTOR	Generally the Displacement Power Factor, but this depends on the device measuring the circuits.
LOAD	Percentage of the Breaker size current that is flowing through the circuit.

**NOTE:** If the entry name requested in the selector does not match a name above and a point under that circuit exists with the same name, the value of the point will be returned.

**Table 5.11 Tenant Entries**

Entry Name	Description
NAME	Name of the tenant
TENANT_ID	The tenant ID
LOCATIONS	Collection of sites associated to circuits linked to this tenant represented by siteName
CIRCUITS	Collection of circuits linked to this tenant represented by siteName/deviceName/circuitName
CIRCUIT_COUNT	Number of circuits linked to this tenant
RACKS	Collection of racks associated to circuits linked to this tenant represented by siteName/rackName
RACK_COUNT	Number of racks associated to circuits linked to this tenant
AMPS_SUM	The sum of circuit total Amps values
AMPS_ALLOCATED	The allocated Amps value
AMPS_USAGE	The percent Amps usage value
KVA_SUM	The sum of circuit total KVA values
KVA_ALLOCATED	The allocated KVA value
KVA_USAGE	The percent KVA usage value
KW_SUM	The sum of circuit total KW values
KW_ALLOCATED	The allocated KW value
KW_USAGE	The percent KW usage value
KWH_NET	The KWH Net Consumption month to date value

**Table 5.11 Tenant Entries (continued)**

Entry Name	Description
LOAD_MAX	The max circuit load value
LOAD_MIN	The min circuit load value
LOAD_AVG	The average circuit load value
CONTACT_NAME	The tenant contact name
CONTACT_PHONE	The tenant contact phone number
CONTACT_EMAIL	The tenant contact email address
ADDRESS_LINE_1	The tenant contact address line 1
ADDRESS_LINE_2	The tenant contact address line 2
ADDRESS_LINE_3	The tenant contact address line 3
KPI1	The values for the point assigned as KPI 1
KPI2	The values for the point assigned as KPI 2
KPI3	The values for the point assigned as KPI 3

**NOTE:** If the entry name requested in the selector does not match a name above and a point under that group exists with the same name, the value of the point will be returned.

**Table 5.12 Point Entries**

Entry Name	Description
NAME	Name of the point.
TYPE	<i>NUMERIC, BOOLEAN, ENUM, STRING</i>
VALUE	Value of the point.
UNITS	Unit facet.
STATUS	Status.
LOCATION	Site name.
DEVICE	Device, if this point is under a device represented by <i>deviceName</i> .
RACK	Rack, if this point is under a rack represented by <i>rackName</i> .
GROUP	Group, if this point is under a group represented by group path.
CIRCUIT	Circuit, if this point is under a circuit represented by <i>circuitName</i> .
TENANT_NAME	Tenant if this point is under a tenant represented by <i>tenantName</i>
TENANT_ID	Tenant if this point is under a tenant represented by <i>tenantId</i>
BOOLEAN_STATUS	<i>On, Off, Alert, Normal</i> .
ALARM_ENABLED	<i>True</i> , if the alarm is enabled.
ALARM_VALUES	Collection of values that cause alarm conditions for <i>BOOLEAN, ENUM and STRING</i> point types.
WARNING_LOW_LIMIT_ENABLED	<i>True</i> , if a numeric warning low threshold is enabled.
WARNING_HIGH_LIMIT_ENABLED	<i>True</i> , if a numeric warning high threshold is enabled.
ALARM_LOW_LIMIT_ENABLED	<i>True</i> , if a numeric alarm low threshold is enabled.

**Table 5.12 Point Entries (continued)**

Entry Name	Description
ALARM_HIGH_LIMIT_ENABLED	<i>True</i> , if a numeric alarm high threshold is enabled.
WARNING_LOW_LIMIT	Numeric warning low threshold value.
WARNING_HIGH_LIMIT	Numeric warning high threshold value.
ALARM_LOW_LIMIT	Numeric alarm low threshold value.
ALARM_HIGH_LIMIT	Numeric alarm high threshold value.
WARNING_ALARM_CLASS	Class used for warning notifications.
ALARM_CLASS	Class used for alarm notifications.
WARNING_TIME_DELAY	Duration in milliseconds the point must be in the warning threshold before triggering a warning.
ALARM_TIME_DELAY	Duration in milliseconds the point must be in the alarm threshold before triggering an alarm.
WARNING_DEADBAND	Value that the point must change by before triggering a warning.
ALARM_DEADBAND	Value that the point must change by before triggering an alarm.
WARNING_MESSAGE	Message sent in the notification.
ALARM_MESSAGE	Message sent in the notification.
WARNING_NORMAL_MESSAGE	Message sent in the return to normal notification.
ALARM_NORMAL_MESSAGE	Message sent in the return to normal notification.
WARNING_INSTRUCTIONS	Instructions sent in the notification.
ALARM_INSTRUCTIONS	Instructions sent in the notification.
HISTORY_ENABLED	<i>True</i> , if history is enabled.
HISTORY_FORCE_INTERVAL	Minimum time in milliseconds before recording a record even if the value has not changed.
HISTORY_CHANGE_TOLERANCE	Value that the point must change by before recording a history record.
HISTORY_RECORD_CAPACITY	Maximum number of historical records stored for the point.
HISTORY_STATE	<i>ENABLED, DISABLED, or FAULT.</i>
HISTORY_FAULT_CAUSE	Message, if state is <i>FAULT</i> .

**Table 5.13 Maintenance Entries**

Entry Name	Description
NAME	Name of the maintenance event.
STATUS	Status of the event.
DEVICES	Collection of devices or racks associated to the event represented by <i>siteName/deviceName</i> or <i>siteName/rackName</i> .
SCHEDULED_START	<i>ISO 8601 datetime.</i>
ACTUAL_START	<i>ISO 8601 datetime.</i>
SCHEDULED_END	<i>ISO 8601 datetime.</i>
ACTUAL_END	<i>ISO 8601 datetime.</i>
SCHEDULED_DURATION	Duration in milliseconds.
ACTUAL_DURATION	Duration in milliseconds.

**Table 5.14 Alarm Entries**

Entry Name	Description
DATE_TIME	<i>ISO 8601 datetime</i> of the alarm.
SOURCE	The source path of the alarm.
ACK_DATE_TIME	<i>ISO 8601 datetime.</i>
RETURN_DATE_TIME	<i>ISO 8601 datetime.</i>
DURATION	Duration in milliseconds.
TIME_TO_ACK	Duration in milliseconds.
DEVICE	Device, if associated to alarm represented by <i>deviceName</i> .
RACK	Rack, if associated to alarm represented by <i>rackName</i> .
GROUP	Group, if associated to alarm represented by group path.
LOCATION	Site, if associated to alarm represented by <i>siteName</i> .
CIRCUIT	Circuit, if associated to alarm represented by <i>circuitName</i> .
TENANT_NAME	Tenant if associated to alarm represented by <i>tenantName</i>
TENANT_ID	Tenant if associated to alarm represented by <i>tenantId</i>
POINT	Point, if associated to alarm represented by <i>pointName</i> .
ALARM_TYPE	Warning or Alarm.
ALARM_VALUE	The value recorded for the alarm.
ACKED_USER	If acknowledged, the username of user who acknowledged the alarm.

## 5.5 Data Set Query Examples

Retrieve data sets for a specific category

**Method:** *GET*

**URL:** */rest/v1/data-sets/DEVICE*

**Content-type:** *application/json*

### Data Sets Entity

```
{
  "type": String,
  "totalCount": Integer,
  "dataSets": [
    {
      "category": String,
      "categoryId": String,
      "displayName": String,
      "path": String,
      "links": {
        "rel": String,
        "href": String
      }
      "entries": [
        {
          "name": String,
          "label": String,
          "values": [
            "value": String,
            "timestamp": ISO 8601 datetime
          ]
        }
      ]
    }
  ]
}
```

**NOTE:** Each category of data set adds additional fields to allow for more fine-grained control over querying that category of data.

### Example HTTP Request

Data sets containing the status entry for devices with names that start with "rpp".

```
$.ajax({
  method : "GET",
  url: "/rest/v1/data-sets/DEVICE?$filter=startswith
(deviceName,'rpp')&ent=Status,STATUS",
  dataType: "json"
});
```

## Example HTTP Response

200 OK

```
{
  "type": "dataSets",
  "totalCount": 1,
  "dataSets": [
    {
      "category": "DEVICE",
      "categoryId": "7cf39723-2c80-4c12-a8df-482973a26c31",
      "displayName": "rpp-1A",
      "deviceName": "rpp-1A",
      "siteName": "CO",
      "path": "CO/rpp-1A",
      "links": [],
      "entries": [
        {
          "name": "STATUS",
          "label": "Status",
          "values": [
            {
              "value": "NORMAL",
              "timestamp": "2015-10-21T16:57:23Z"
            }
          ]
        }
      ]
    }
  ]
}
```

## Example HTTP Request

Data sets containing the Amps entry for the device "CO/rPDU-81A" with a monthly average rollup given a custom timeframe of two months.

```
$.ajax({
  method : "GET",
  url: "/rest/v1/data-sets/DEVICE?$filter=path eq 'CO/rPDU-81A'&ent=Amps,Amps,MONTHLY,AVG,USER_DEFINED,2015-09-01T06:00:00Z,2015-11-01T06:00:00Z",
  dataType: "json"
});
```

## Example HTTP Response

200 OK

```
{
  "type": "dataSets",
  "totalCount": 1,
```

```

"dataSets": [
  {
    "category": "DEVICE",
    "categoryId": "f9585573-afa5-43d9-ba82-2544fd540559",
    "displayName": "rPDU-81A",
    "deviceName": "rPDU-81A",
    "siteName": "CO",
    "path": "CO/rPDU-81A",
    "links": [],
    "entries": [
      {
        "name": "Amps",
        "label": "Amps",
        "units": "AMPS",
        "status": "NORMAL",
        "pointId": "4fd59457-4152-491a-a2b5-a085c65793a9",
        "alarmLowLimit": "-Infinity",
        "alarmHighLimit": "Infinity",
        "warnLowLimit": "-Infinity",
        "warnHighLimit": "Infinity",
        "values": [
          {
            "value": "4",
            "timestamp": "2015-10-01T06:00:00Z"
          },
          {
            "value": "4",
            "timestamp": "2015-11-01T06:00:00Z"
          }
        ]
      }
    ]
  }
]
}

```

### Example HTTP Request

Requesting the delta temperature of a CRAC unit by requesting the inlet temperature, the return temperature and a calculated delta temperature using the *SUB()* function to subtract the outlet temperature from the return temperature.

```

$.ajax({
  method: "GET",
  url: "/rest/v1/data-sets/DEVICE?${filter}=path eq 'CO/CRAC-
A1'&ent=Return+Temperature,Return+Temperature,ALL,AVG,NOW,, , ;Temperature,Temperature,ALL
,AVG,NOW,, , , ;Delta+Temp,%5B+Calculated+Value+%5D,ALL,AVG,NOW,, , , SUB
(Return+Temperature, Temperature)",
  dataType: "json"
});

```

## Example HTTP Response

200 OK

```

{
  "type": "dataSets",
  "totalCount": 1,
  "dataSets": [
    {
      "category": "DEVICE",
      "categoryId": "7cf39723-2c80-4c12-a8df-482973a26c31",
      "displayName": "CRAC-A1",
      "deviceName": "CRAC-A1",
      "siteName": "CO",
      "path": "CO/CRAC-A1",
      "links": [],
      "entries": [
        {
          "name": "Return Temperature",
          "label": "ent=Return Temperature",
          "units": "FAHRENHEIT ",
          "status": "NORMAL",
          "pointId": "32240eb5-1a9a-4cd8-a5de-2d6e1da952f0",
          "alarmLowLimit": "55.0",
          "alarmHighLimit": "90.0",
          "warnLowLimit": "60.0",
          "warnHighLimit": "85.0",
          "values": [
            {
              "value": "77",
              "timestamp": "2018-09-07T18:50:33Z"
            }
          ]
        }
      ],
      {
        "name": "Temperature",
        "label": "Temperature",
        "units": "FAHRENHEIT ",
        "status": "NORMAL",
        "pointId": "f5f089de-1a15-432c-914d-737ce65aae36",
        "alarmLowLimit": "55.0",
        "alarmHighLimit": "95.0",
        "warnLowLimit": "60.0",
        "warnHighLimit": "85.0",
        "values": [
          {
            "value": "64.0",
            "timestamp": "2018-09-07T18:50:33Z"
          }
        ]
      }
    ],
    {
      "name": "[ Calculated Value ]",
      "label": "Delta Temp",
      "values": [
        {
          "value": "13.0",
          "timestamp": "2018-09-07T18:50:33Z"
        }
      ]
    }
  ]
}

```

```
}  
  ]  
    }  
      ]  
        }  
          ]
```

## 6 Analytics Metadata

Allows retrieving metadata that describes the data in the system and how to work with it. It is possible to retrieve a definition of the information available for any data set category, category instance, time frame, rollup or rollup function.

**Table 6.1 Alarm Entries**

Method	URL	Content Type	Description
GET	<code>/rest/v1/meta/dataset/categories/category</code>	application/json	Get metadata about a data set category.

### 6.1 Query for Data Set Category Metadata

Retrieve metadata about a category. The example below retrieves metadata about the *DEVICE* category.

**Method:** *GET*

**URL:** `/rest/v1/meta/dataset/categories/DEVICE`

**Content-type:** *application/json*

#### Data Set Category Metadata Entity

```
{
  "category": String,
  "display": String,
  "entries": [
    {
      "name": String,
      "isHistorical": Boolean,
      "isNumeric": Boolean,
      "isCollection": Boolean,
      "partialMatch": Boolean
    }
  ]
}
```

The metadata will contain information for all possible entries for the given category. *isHistorical* indicates that historical timeframes can be used with this entry. *isNumeric* indicates that the values for this entry are numerical. *isCollection* indicates that each value returned is not historical but, rather represents an item in a collection of values. *partialMatch* indicates that the entry is only available for a subset of data sets matched by the query.

**NOTE:** The `selectors` field provides information about the legacy selectors that are now deprecated in favor of the new filter system.

### Example HTTP Request

*//Get metadata about the DEVICE category for devices with names starting with "rpp"*

```
$.ajax({
  method: "GET",
  url: "/rest/v1/meta/dataset/categories/DEVICE?$filter=startswith(deviceName,'rpp')",
  dataType: "json"
});
```

### Example HTTP Response

200 OK

```
{
  "category": "DEVICE",
  "display": "Device",
  "entries": [
    {
      "name": "NAME",
      "type": "NON_NUMERIC",
      "isHistorical": false,
      "isNumeric": false,
      "isCollection": false,
      "partialMatch": false
    },
    {
      "name": "STATUS",
      "type": "NON_NUMERIC",
      "isHistorical": false,
      "isNumeric": false,
      "isCollection": false,
      "partialMatch": false
    }
  ]
}
```

### **Connect with Vertiv on Social Media**



<https://www.facebook.com/vertiv/>



<https://www.instagram.com/vertiv/>



<https://www.linkedin.com/company/vertiv/>



<https://www.twitter.com/Vertiv/>



---

Vertiv.com | Vertiv Headquarters, 1050 Dearborn Drive, Columbus, OH, 43085, USA

© 2022 Vertiv Group Corp. All rights reserved. Vertiv™ and the Vertiv logo are trademarks or registered trademarks of Vertiv Group Corp. All other names and logos referred to are trade names, trademarks or registered trademarks of their respective owners. While every precaution has been taken to ensure accuracy and completeness here, Vertiv Group Corp. assumes no responsibility, and disclaims all liability, for damages resulting from use of this information or for any errors or omissions. Specifications, rebates and other promotional offers are subject to change at Vertiv's sole discretion upon notice.

SL-70596\_REV4\_01-22